Introduzione all'informatica

Lezione 1 di Fondamenti di informatica

Docente: Giuseppe Scollo

Università di Catania
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica, I livello, AA 2009-10

Indice

1. Introduzione all'informatica
2. di che tratta l'informatica?
3. algoritmi! (e dintorni)
4. esempi di comuni algoritmi
5. definizioni di algoritmo
6. addizione Maya: algoritmi "manuali"
7. terminazione di algoritmi: il caso 3x+1
8. una definizione più ampia di algoritmo
9. algoritmi e calcolo: precedenti storici
10. precursori nella rivoluzione industriale
11. la macchina analitica di Babbage
12. calcolatori elettromeccanici e ... teorici
13. calcolatori elettronici, Von Neumann
14. informatica contemporanea
15. temi per ulteriori approfondimenti

di che tratta l'informatica?

primo dovere di una disciplina scientifica è la **definizione dell'oggetto di studio**

probabilmente a causa della pervasiva presenza di applicazioni dell'informatica nella società odierna, convinzioni erronee e luoghi comuni sull'oggetto proprio di questa scienza sono diffusi

ad esempio (e non si tratta degli equivoci più gravi), spesso si **identifica** l'informatica con lo studio

dei calcolatori, o anche

di tecniche di programmazione dei calcolatori, o persino

di usi e applicazioni di calcolatori e programmi

conoscenze di tal sorta sono rilevanti all'oggetto dell'informatica, e professionalmente utili, ma

altra è l'"**essenza**" dell'informatica ... di che si tratta?

algoritmi! (e dintorni)

N. Gibbs e A. Tucker (1986) individuano nello **studio di algoritmi** la caratteristica centrale dell'informatica, che comprende le loro:

proprietà formali e matematiche
realizzazioni hardware
realizzazioni linguistiche
applicazioni

questa caratterizzazione ha un ampio spettro di competenze... che c'è al fondo?

occorre una **definizione** del termine "**algoritmo**" [da dizionario:]

Algoritmo *s.m. (dal nome del matematico persiano al-Khwarizmi), sistema di regole e procedure di calcolo ben definite che portano alla soluzione di un problema con un numero finito di operazioni*

come spesso accade con le definizioni, ne occorrono altre...

il fatto un po' preoccupante è però che una "procedura di calcolo" altro non è che un (caso particolare di) algoritmo (con altro nome)...

esempi di comuni algoritmi

un algoritmo per la **cottura dell'uovo alla *coque***

1. immergi completamente l'uovo in un pentolino con acqua fredda
2. se l'uovo galleggia sostituiscilo con uno più fresco e ricomincia dal passo 1
3. poni il pentolino sul fornello
4. accendi una fiamma vivace sotto il pentolino
5. attendi fino all'ebollizione
6. modera la fiamma sotto il pentolino
7. attendi 100 secondi circa
8. spegni la fiamma sotto il pentolino
9. blocca rapidamente la cottura temperando con acqua fredda

un algoritmo per la **conversione da Celsius a Fahreneit**

1. moltiplica il valore della temperatura in gradi Celsius per 9
2. dividi per 5 il risultato del passo precedente
3. aggiungi 32 al risultato del passo precedente

definizioni più formali di algoritmo

il carattere fondamentale del concetto di algoritmo in informatica è riconoscibile nella definizione di quest'ultima come

> ***scienza della risoluzione di problemi mediante algoritmi***

l'utilità pratica di algoritmi ben formalizzati deriva dal fatto che essi permettono di **automatizzare** la risoluzione di (classi di) problemi

> una definizione più formale del concetto di algoritmo serve invece alla nostra comprensione dell'esatta natura dell'oggetto di questa scienza

il testo (Schneider & Gersting, 2007) propone la seguente:

> *un insieme ordinato di operazioni non ambigue ed effettivamente computabili che, quando eseguito, produce un risultato e si arresta in un tempo finito*

come gli stessi autori notano, la definizione è piuttosto stringente... forse un po' troppo?

1. sebbene la definizione non lo implichi, gli autori intendono un ordinamento totale delle operazioni da eseguire, ovvero una **unica sequenza di esecuzione** (per un dato input); vediamo appresso un esempio che solleva qualche dubbio in proposito;
2. il requisito di arresto in ogni caso è inoltre piuttosto critico, come vedremo

addizione Maya: algoritmi "manuali"

il testo citato illustra la definizione di algoritmo con un esempio familiare dai banchi di scuola elementare: l'addizione di due numeri; lo si confronti con quello analogo, ma con una diversa scelta di simboli per le cifre, mutuata dalla cultura Maya (per semplicità qui in base 10: la rappresentazione Maya dei numeri è in base 20):
le cifre diverse da zero sono costruite mettendo assieme costituenti di due tipi:

- • : vale 1 (lo chiamiamo "unità")
- —— : vale 5 (lo chiamiamo "cinquina")

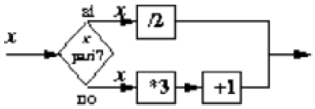
in base 10, un cifra non nulla è costituita da al più 4 unità e al più 1 cinquina
ecco un procedimento per sommare due (o più) numeri in tale rappresentazione:

1. allineare a destra le sequenze di cifre che rappresentano i numeri da sommare
2. raggruppare in fondo a ciascuna colonna i costituenti delle cifre di quella colonna
3. trasformare la raccolta ottenuta da ciascuna colonna, ed eventualmente altre di maggiore significatività generate dal procedimento, applicando in qualsiasi ordine, in ciascuna posizione, le due regole seguenti, fintantoché applicabili:
   se presenti, rimpiazza 5 unità con una cinquina
   se presenti, rimpiazza 2 cinquine nella posizione data con una unità nella successiva posizione più significativa (a sinistra)

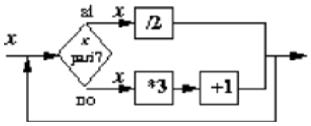
terminazione di algoritmi: il caso 3x+1

sia $f : N_+ \to N_+$ la funzione definita sui numeri naturali positivi (0 escluso) dalla regola:
se n è pari, allora $f\,n = n/2$, altrimenti $f\,n = 3n + 1$

un algoritmo di calcolo per questa funzione può essere descritto da un **diagramma a blocchi**



con un piccolo ritocco al diagramma a blocchi, lo trasformiamo in quello di un **sistema dinamico**, che, avviato da un input x, genera la sequenza infinita delle applicazioni iterate di $f : f\,x,\ f\,f\,x,\ f\,f\,f\,x,\ \ldots$



sia $f^k x$ il k-simo elemento della sequenza; **ritardo** di x ≡ minimo k tale che $f^k x = 1$
Problema: tale minimo esiste per qualsiasi x ? Risposta: forse ...
dal diagramma a blocchi del sistema dinamico si ottiene facilmente quello di un algoritmo per il calcolo del ritardo, ma...
stando alla definizione di algoritmo vista sopra, **sarebbe** un algoritmo?
risposta: come sopra :(

una definizione più ampia di algoritmo

ecco una definizione informale meno esigente di quella citata, tale da permettere di annoverare fra gli algoritmi i procedimenti visti nei due esempi precedenti:

un insieme ordinato di passi eseguibili e non ambigui

dove è inteso che

- l'ordine non sia necessariamente sequenziale, ovvero sia **parziale**
- esistano uno o più **esecutori** in grado di **interpretare** i passi (purché non ambigui) ed **effettivamente** eseguirli rispettando l'ordine prescritto

la definizione **non** prescrive che l'esecuzione **termini**

la non terminazione di un algoritmo **non** è necessariamente sintomo di un errore di progettazione o di realizzazione dell'algoritmo

- ad esempio, algoritmi di allocazione e rilascio di risorse, o di dispaccio di richieste di servizio, tipicamente devono esaminare ciclicamente richieste in arrivo da un insieme finito di canali in ingresso, per servirle secondo un disciplina predefinita; la presenza di un ciclo infinito è una caratteristica inerente a questi algoritmi

algoritmi e calcolo: precedenti storici

un rapida escursione introduttiva:

- etimologia di "**calcolo**"
- antichi algoritmi e strumenti di calcolo: ***abaci***
- il ***Liber abaci*** di Leonardo Pisano (**Fibonacci**)
- le calcolatrici del XVII secolo: ***Pascalina, Stepped Reckoner (Leibniz)***
- Leibniz: ***"calculemus!"***
- rivoluzione industriale: telai **Jacquard**, logaritmi di **Prony**
- XIX secolo: l'***Analytical engine*** di **Babbage**
- **Boole, De Morgan, Pierce, Schröder, Cantor, Frege, ...**
- XX secolo, il problema dei fondamenti della matematica:
  - paradossi, programma di **Hilbert**, teorema di **Gödel**
- i primi calcolatori sono "virtuali": **Post, Turing, Church**
- **Von Neumann** : architettura del calcolatore "reale"
- anni '50: l'Informatica dalla preistoria alla storia

precursori nella rivoluzione industriale

l'idea di J.-M. Jacquard (1804): introdurre nei telai delle schede di cartone forato; ad ogni scheda corrispondeva un preciso disegno, formato dai fori

telaio Jacquard
fonte: Wikimedia Commons

lettura delle schede: le file di aghi possono attraversarle solo dove ci sono i fori

i fili corrispondenti alla trama programmata sono sollevati automaticamente, permettendone il passaggio

un addetto basta ad operare con questo telaio, invece di tre ...

analogie con i moderni calcolatori:

macchina a programma = sequenza di istruzioni

rappresentazione binaria del programma

per maggiori dettagli sul funzionamento del telaio Jacquard: v. sito del Museo del Tessile di Chieri

la macchina analitica di Babbage

la prima invenzione di Charles Babbage è la **macchina differenziale** (1823): estende le idee di Pascal e Leibniz, può operare fino a 6 cifre significative

questa fu la sola che riuscì a costruire ...

il progetto di costruirne una con precisione a 20 cifre è stato realizzato... nel 1991, dopo sei anni di lavoro, dal London Museum of Science

la successiva **macchina analitica** non è però una mera (per quanto sofisticata) calcolatrice:

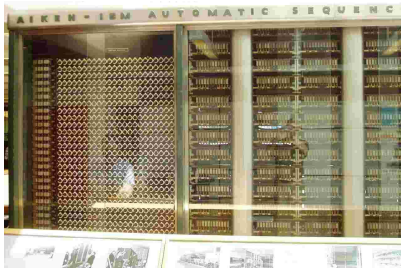
la sua architettura è straordinariamente simile a quella del moderno calcolatore! v.: Sketch of the Analytical Engine di L. F. Menabrea, tradotto e corredato di ricche note da Ada Lovelace, "*prima programmatrice della storia*"

Ada Augusta Byron, contessa di Lovelace, 1838
fonte: Wikimedia Commons

calcolatori elettromeccanici e ... teorici

XX secolo: elaborazione automatica di dati su larga scala, motivazioni:

civili: censimento, grandi aziende, ... : macchine di **Hollerith**
— non a caso, in francese "calcolatore" si traduce in "ordinateur" —
militari: codici crittografici, calcoli balistici, ...
enormi macchine da calcolo elettromeccaniche, come ad es. Mark I



Porzione del calcolatore Mark I
fonte: Wikimedia Commons

non meno importanti sono gli **sviluppi teorici** dei primi decenni, che negli anni '30 producono:

modelli concettuali, simbolici, di macchine da calcolo: **Post, Turing, Church**
un concetto generale di **calcolabilità** (di una funzione): **Tesi di Church-Turing**
un **risultato fondamentale** per l'Informatica: l'indecidibilità dell'arresto della macchina di Turing universale

calcolatori elettronici, Von Neumann

negli anni '40, grazie alla disponibilità delle prime tecnologie elettroniche, prende definitivamente avvento la **rappresentazione binaria dell'informazione**

non è una novità: ad es., il codice Morse aveva fatto la sua comparsa con l'invenzione del telegrafo, nel secolo precedente

spesso si riporta quale primo esemplare di calcolatore elettronico l'ENIAC, costruito da Eckert e Mauchly (U. of Pennsylvania, 1946), ma la questione è controversa

v. ad es. la storia del calcolatore di Atanasoff e Berry (ABC)

altri predecessori di ENIAC sono **Colossus** (A. Turing, U. of Manchester), costruito per scopi bellici e coperto dal segreto militare, ricostruito al National Museum of Computing (1994—2004), e le macchine programmabili Z1—Z4 (K. Zuse, Berlin)

l'idea di Von Neumann (che aveva partecipato al progetto ENIAC): rappresentare e memorizzare il **programma** di calcolo così come si rappresentano e memorizzano i **dati** su cui opera

l'idea si era già manifestata nel concetto di Macchina di Turing Universale

con Von Neumann, l'idea prende "corpo tecnologico": nasce il calcolatore ***general-purpose* a programma memorizzato**

primo esemplare della specie è l'EDVAC (1951), UNIVAC I ne è la versione commerciale, l'EDSAC (M. Wilkes, U. of Cambridge) è dello stesso periodo

informatica contemporanea

sviluppi salienti:

anni '50:
- linguaggi di programmazione simbolici, ***assembler***
- impiego di transistor, memorie a nuclei, dischi magnetici
- primi linguaggi di programmazione ad alto livello

anni '60 e '70 (prima metà):
- sistemi operativi, ***time-sharing***, programmazione strutturata
- impiego di circuiti integrati, primi minicalcolatori
- industria del software, standard di compatibilità fra sistemi

anni '70 (seconda metà) e '80:
- programmazione di sistema (Unix), dichiarativa, ad oggetti
- impiego di circuiti integrati LSI, VLSI, ULSI
- microcalcolatori, calcolatori personali, interfacce grafiche (GUI)
- reti di calcolatori, basi di dati

dagli anni '90 ad oggi:
- Internet, World Wide Web, standard per l'industria del software
- architetture parallele, supercalcolatori, enormi capacità di memoria
- calcolatori portatili e palmari, interfacce multimediali
- standard per suoni, immagini e filmati digitali, informatica pervasiva

temi per ulteriori approfondimenti

1. **Precisione della definizione di "algoritmo"**
   Né la definizione di algoritmo qui proposta, né le sue varianti più comuni, precisano se l'ordine a cui si fa riferimento riguardi la **descrizione** dei passi dell'algoritmo o piuttosto la loro **esecuzione.** Riflettere sulle differenze fra le due interpretazioni della definizione che ne conseguono, e possibilmente proporre una definizione migliore.
2. **Approfondimenti storici su algoritmi e calcolo**
   In ciascuno dei 12 punti qui proposti (e in altri "in mezzo") fra i precedenti storici del rapporto fra algoritmi e calcolo, si individuano facilmente uno o più temi suscettibili di approfondimento storico; sceglierne uno di personale interesse, reperire fonti attendibili di studio sul tema e produrre una sintesi documentata dello studio svolto.
3. **Algoritmi paralleli e griglie computazionali**
   Molti problemi possono essere risolti da algoritmi eseguibili da più agenti di calcolo che operano contemporaneamente, spesso ciascuno su una parte dei dati in ingresso (ad esempio, l'algoritmo per l'addizione Maya qui descritto ha questa proprietà, dove ciascun esecutore opera sulle cifre di una sola colonna); le **griglie computazionali** sono reti di calcolatori atte all'esecuzione di tali algoritmi, e il termine ***grid computing*** designa questo tipo di elaborazione; raccogliere documentazione su queste reti e/o su algoritmi eseguiti su griglie e produrne una sintesi, corredata di riferimenti alle fonti.
4. **Altri temi proposti dal testo** (Schneider & Gersting, 2007): p.32, Esercizi 10,12,15.